

BAB II

TINJAUAN PUSTAKA

A. Tinjauan Pustaka

SMS gateway merupakan mekanisme mengirim dan menerima pesan singkat berupa teks melalui sebuah komputer yang terhubung ke telepon selular atau *modem GSM* melalui *serial port*, *Infrared Data Association (IrDA) port* maupun *bluetooth*. Dimana telepon selular maupun *modem GSM* berfungsi sebagai *modem*. Arsitektur ini disebut *independent service*. Arsitektur lain untuk menghubungkan antara penerima dan penyedia informasi melalui *SMS* yaitu *dependent service*, dimana komputer yang berfungsi sebagai *server gateway* terhubung secara langsung ke server operator selular melalui internet (Pramsane & Sanjaya, 2006).

Dengan aplikasi *SMS gateway*, *SMS* dapat dimanfaatkan untuk keperluan yang lebih luas dalam menyediakan informasi sejenis bagi banyak orang sesuai permintaan dengan format tertentu secara otomatis. Beberapa penelitian sebelumnya telah membahas implementasi *SMS gateway* terutama dalam dunia bisnis antara lain : *SMS gateway* untuk layanan informasi kejuaran sepak bola Liga Inggris yang menggunakan bahasa pemrograman Visual Basic dengan memanfaatkan AT Command (Cahyo dkk, 2006), implementasi *SMS gateway* pada *Multi Level Marketing (MLM)* untuk memudahkan proses registrasi anggota (distributor) baru, penjualan produk, memberi informasi bonus dan informasi penting berupa alert kepada anggota (Mee & Selamat, 2007).

Beberapa penelitian lain membahas tentang implementasi *SMS gateway* pada dunia pendidikan, antara lain : penambahan fitur *SMS gateway* pada *Student Record Retrieval System* yang memudahkan siswa dalam mengakses informasi terutama dari gurunya secara *up to date* (Muhamadi dkk, 2009), *SMS gateway* untuk layanan administrasi pada perpustakaan dalam memberi informasi peringatan kepada peminjam mengenai jatuh tempo pengembalian buku, informasi ke penerbit/supplier mengenai penerimaan buku, kerusakan atau kehilangan, pengumuman dan lain-lain (Manoj dkk, 2007; Vimal and Chitra, 2008), pemanfaatan *SMS* untuk interaksi tanya jawab dan jajak pendapat di dalam ruang kelas. Penerapan sistem ini bertujuan untuk meningkatkan kualitas dan kuantitas umpan balik dari siswa, terutama siswa yang malu bertanya secara langsung di depan banyak orang (Scornavacca dkk, 2007). Pemanfaatan *SMS* pada *m-Learning*, dimana siswa dapat menjawab pertanyaan singkat dengan berbagai tipe dan sistem secara otomatis akan mengevaluasi jawaban pertanyaan tersebut dengan teknik pencocokan, dan memberikan umpan balik serta penilaian (Balasundaram and Ramadoss, 2007).

Pada bidang pemerintahan, model *e-government* berbasis *SMS* adalah sebuah model tahap pembangunan bagi pemerintah yang ingin fokus memberikan pelayanan publik dengan menggunakan teknologi *SMS*. Model ini mengusulkan bahwa pemerintah dapat mengembangkan layanan mereka yang berbasis *SMS* melalui lima level yaitu : pemberitahuan, presentasi, komunikasi, transaksi dan integrasi. Setiap level menggambarkan layanan *e-government* yang dapat disampaikan melalui sistem *SMS*.

Pada level pemberitahuan, layanan *SMS* bersifat satu arah dari pemerintah ke masyarakat. Layanan pada level ini menggunakan tipe pengiriman *push SMS* yang dapat diterapkan pada beberapa departemen seperti departemen meteorologi dalam menyediakan laporan cuaca dan peringatan bencana, departemen transportasi dalam menyediakan informasi keterlambatan kedatangan kereta api, informasi penutupan beberapa jalan dan pemberitahuan tentang suatu *event* atau peraturan baru. Pada departemen pajak, sistem secara otomatis mengirimkan pesan peringatan kepada warga tentang batas waktu pembayaran pajak. Dan pada departemen tenaga kerja, sistem secara otomatis mengirimkan informasi lowongan kerja kepada warga yang terdaftar sebagai pengangguran. Teknologi yang dibutuhkan pada level ini meliputi basis data klien/warga yang berisi data diri dan nomor telepon selular mereka, *SMS Broadcasting* yaitu sistem *SMS point to point* yang memungkinkan pengiriman pesan serempak ke banyak penerima dan *Cell Broadcast Service* yaitu teknologi untuk mengirimkan pesan-pesan yang bersifat darurat ke semua *mobile stations* pada area khusus, tanpa memilih penerima pesan. Teknologi *Cell Broadcast Service* memiliki beberapa kelebihan yaitu probabilitas penerimaan pesan yang tidak lengkap sangat rendah karena pesan dikirim berdasarkan periode waktu tertentu dan *mobile stations* hanya dapat menerima pesan selanjutnya jika pesan sebelumnya telah diterima. Keuntungan lain yaitu data dapat dikirim sebagai binary data atau *American Standard Code for Information Interchange (ASCII) text* dengan panjang hingga 15 halaman, masing-masing halaman dapat mencapai panjang hingga 93 karakter yang dapat di-*broadcast* sebagai satu pesan.

Pada level presentasi, pengiriman *SMS* dapat dilakukan melalui dua mekanisme yaitu *push-based* dan *pull-based*. *Pull-based* bersifat dua arah berdasarkan permintaan dari warga. Layanan ini merupakan penambahan pada level pemberitahuan. Pada level komunikasi, terdapat penambahan fasilitas bagi warga untuk bertanya dan memperoleh jawaban singkat melalui *SMS* dalam waktu yang singkat. Format pertanyaan dan jawaban standar telah disimpan dalam basis data. Apabila ada pertanyaan baru dan jawabannya yang tidak tersimpan dalam basis data, maka pertanyaan tersebut akan diforward ke nomor telepon selular petugas yang mengatur layanan tersebut, selanjutnya petugas akan menjawab pertanyaan tersebut dan menginputkan pertanyaan dan jawabannya sebagai format standar ke basis data. Sedangkan pertanyaan yang membutuhkan jawaban detail, dapat dijawab melalui surat, email, telepon atau bertemu secara langsung.

Level selanjutnya yaitu level transaksi, yang memungkinkan warga untuk melakukan transaksi melalui *SMS*. Pada level ini, autentikasi, verifikasi dan *issue security* perlu diperhatikan. Dan level terakhir adalah level yang mengintegrasikan setiap sistem berbasis *SMS* dengan keseluruhan sistem *e-government* lainnya, termasuk sistem *e-government* berbasis web (Susanto and Goodwin, 2006). Selanjutnya, Susanto dkk (2008) mengusulkan penambahan 1 tingkat pada level terendah dalam pengembangan sistem *e-government* berbasis *SMS* yaitu level mendengarkan yang bersifat satu arah dari warga kepada pemerintah berupa kritik dan saran. Setiap level memiliki fungsi masing-masing dan dapat saling melengkapi serta dapat menggantikan peran *e-government* berbasis web.

SMS gateway juga dapat dimanfaatkan untuk mengontrol keamanan rumah. *Home Appliance Control System (HACS)* adalah sistem yang mengusulkan dua sub-sistem. Subsistem kontrol Appliance memungkinkan pengguna untuk mengontrol jarak jauh peralatan rumah mereka, sedangkan subsistem peringatan keamanan menyediakan pemantauan keamanan jarak jauh. Sistem ini cukup mampu untuk mengarahkan pengguna melalui *SMS* dari sejumlah ruang tertentu untuk mengubah kondisi alat rumah sesuai dengan kebutuhan pengguna. Aspek kedua adalah peringatan keamanan yang dicapai memungkinkan generate *SMS* secara otomatis sehingga dapat memperingatkan pengguna terhadap risiko bahaya yang mungkin terjadi (Khiyal dkk, 2009).

Selain untuk mengontrol keamanan rumah, *SMS gateway* juga dapat diimplementasikan dalam mekanisme penanganan *spam* (Seo dkk, 2007). Penelitian yang dilakukan oleh Seo dkk ini menawarkan metode dimana pengirim email akan menerima informasi rahasia dari *SMS server* pengirim melalui *SMS*. Informasi tersebut berupa *Key session (Ks)* yang digunakan untuk melakukan enkripsi terhadap email yang akan dikirimkan. Proses ini menggunakan informasi telepon selular yang dilaporkan ketika berlangganan ke layanan email. Selanjutnya, email ditandai dengan menggunakan kunci privat dan dienkripsi menggunakan *Ks* yang digenerate dan dikirim oleh *SMS server*. Kemudian email tersebut ditransmisikan ke *mail server* pengirim untuk dikirimkan ke *mail server* penerima melalui *Simple Mail Transfer Protocol (SMTP)*. Pada sisi penerima, email dapat didekripsi menggunakan *Ks* dan *Ks* dapat diperoleh dengan menggunakan kunci privat penerima. Penerima mengotentikasi pengirim menggunakan kunci publik

pengirim. Jika penerima berhasil mengotentikasi pengirim, maka penerima akan menerima pesan teks yang asli. Karena metode yang ditawarkan mengotentikasi pengirim dalam proses transmisi email, maka dapat mencegah email spam.

Dari beberapa penelitian yang telah dibahas, arsitektur yang paling banyak digunakan adalah arsitektur yang bersifat *independent service*. Namun dari kasus-kasus yang dibahas pada penelitian-penelitian tersebut tidak bersifat heterogen, artinya aplikasi yang dibangun merupakan satu kesatuan sistem yang tidak berkaitan dengan sistem yang lainnya dan berada pada *platform* yang sama. Akses basis data oleh *server gateway* juga dilakukan secara langsung ke *server* basis data. Arsitektur ini lebih sederhana, namun jika pengembangan sistem dilakukan secara bertahap dan bersifat fungsional untuk masing-masing bagian, dimana masing-masing bagian ini merupakan sebuah sistem yang independen dan membutuhkan beberapa data yang sama, maka pengembangan sistem yang baru tentu saja harus selaras dengan sistem yang dibuat sebelumnya, khususnya pengelolaan data. Agar tidak tumpang tindih, sistem-sistem tersebut dapat diintegrasikan sehingga dapat saling berbagi pakai data.

Seiring dengan perkembangan teknologi, ada banyak cara untuk melakukan integrasi antar sistem, baik pada *level* data maupun pada *level logic* aplikasi. Pemilihan jenis integrasi dan penggunaan teknologi integrasi dapat disesuaikan dengan kebutuhan. Calvanese dkk menggunakan pendekatan deklaratif untuk mengintegrasikan sumber-sumber data pada *data warehouse* berdasarkan pada formalisme pemodelan konseptual ekspresif yang dilengkapi dengan teknik penalaran. Kerangka kerja yang disajikan dalam penelitian tersebut diterapkan

untuk mengatasi masalah desain *data warehouse* dalam proyek ESPRIT DWQ (Calvanese dkk, 2001).

Dalam penelitian yang dilakukan oleh Julistiono dan Lamidjan (2001) digunakan teknologi *CORBA* untuk membuat aplikasi untuk reservasi hotel dengan sistem *multi-tier*. Proses reservasi dapat dilakukan baik melalui *web browser* maupun program *client* yang dibuat dengan bahasa pemrograman Delphi. Program *client* Delphi melalui *CORBA connection* akan dihubungkan ke aplikasi *server*. Dan melalui *ODBC*, aplikasi *server* akan dihubungkan ke *DBMS SQL Server*. Julistiono dan Lamidjan (2001) menyatakan bahwa :

“penggunaan sistem *multi-tier* dengan *CORBA* ini secara umum memiliki keuntungan selain dapat dikembangkan lebih lanjut, sistem ini juga dirancang dengan sistem *multi database server*, *multi middle server*, dan *multi client* dimana dengan hal ini seluruh sistem dapat terkait menjadi satu. Kelemahan sistem ini adalah struktur sistem *CORBA* yang rumit sehingga sulit dipahami, sedangkan untuk *multi-tier* itu sendiri dibutuhkan suatu prosedur tersendiri untuk menentukan *server* mana yang menjadi pilihan dari *client*”.

Teknologi berbasis komponen seperti *CORBA* dan *RMI* berorientasi pada koneksi dan menggunakan *port* komunikasi yang tidak standart, sebagai alasan terhadap masalah hak akses terkait isu keamanan. Hal ini menjadikan teknologi ini lebih *secure*, namun membutuhkan koneksi yang cepat dan stabil. Teknologi ini juga tidak dapat menangani gangguan jaringan dengan baik, membutuhkan *cost (source)* yang besar untuk menghubungkan kembali ke *remote server* setelah terjadi *crash/corruption* (Shetty and Vadivel, 2009).

Seiring dengan perkembangan internet dan meningkatnya jumlah penggunaan internet menjadikan protokol *HTTP* sebagai media yang banyak digunakan untuk saling berbagi *service*. *Web service* merupakan pendekatan terhadap masalah

komunikasi antar aplikasi yang berbeda *platform, framework* dan tipe data dengan memanfaatkan *HTTP* sebagai media *transport* datanya (Fajar, 2009). *Web service* dapat diartikan sebagai sebuah antar muka (*interface*) yang berbentuk *function/method* berisi sekumpulan *code* yang dapat diakses melalui jaringan, oleh berbagai aplikasi. Dengan menggunakan *web service*; aplikasi dengan sistem operasi dan basis data yang berbeda arsitektur dapat diintegrasikan, sehingga dapat saling berkomunikasi dan bertukar data, dan akan mengurangi biaya integrasi (Zhang dkk, 2006; Kumar, 2011). *Web service* mampu mengintegrasikan aplikasi yang berbeda secara lebih sederhana dan mampu memperbaiki kelemahan dari teknologi sistem terdistribusi sebelumnya. Dengan menggunakan *HTTP* atau *SMTP*, *web service* bisa menembus pengamanan *firewall* suatu organisasi tanpa mengubah konfigurasi *firewall*. *Web service* juga bersifat *loosely-coupled* terhadap *client* (Kuswandi dkk, 2007; Johnstone, 2001).

Ada beberapa jenis *web service* dengan aturannya masing-masing. Dalam mengimplementasikannya, dapat dipilih jenis *web service* yang sesuai dengan kebutuhan performansi, kecepatan, kehandalan, keamanan dan lain-lain. Beberapa penelitian sebelumnya telah membahas tentang implementasi dan kinerja dari beberapa jenis teknologi *web service*, antara lain penelitian yang dilakukan oleh Yoon dan Park (2008), yang membahas tentang berbagai bentuk implementasi dari *Service-oriented Architecture (SOA)* untuk memenuhi kebutuhan dalam beradaptasi terhadap keadaan (adaptasi secara dinamis terhadap perubahan frekuensi jumlah pesan *Extensible Markup Language (XML)* antar berbagai

macam situs/aplikasi) dan mempertahankan *performance* eksekusi (kecepatan dan kelancaran dalam transfer data).

Penelitian lainnya dilakukan oleh Priyambodo (2005) yang membahas tentang *web service* dan mengimplementasikannya untuk membangun *service* yang akan memberikan rekomendasi mengenai sejumlah obyek wisata di Jawa Tengah dan Yogyakarta yang sebaiknya dikunjungi oleh wisatawan dalam rangka wisata budaya Jawa. *Service* tersebut dikembangkan dengan tujuan untuk melihat bagaimana *web service* dapat mengakomodasi atau dimanfaatkan dalam membangun sebuah program aplikasi, memperlihatkan integrasi antar *web service* serta untuk mengetahui apakah *web service* dapat digunakan pada komputer dengan sistem operasi yang berbeda dengan sistem operasi *service provider*. Implementasi sistem ini menggunakan *Simple Object Access Protocol (SOAP)*.

Zhang dkk (2006) melakukan penelitian serupa menggunakan *SOAP* yang diimplementasikan untuk melakukan pertukaran data antara dua buah sistem. Kedua sistem tersebut awalnya dikembangkan pada saat yang bersamaan di dua tempat yang berbeda untuk keperluan yang berbeda. Sistem manajemen permintaan penawaran dan masa tunggu proyek konstruksi (sistem A) yang berfungsi untuk memajemen proses permintaan penawaran proyek konstruksi secara keseluruhan dan menunggu registrasi proyek, pemanggilan untuk registrasi penawaran, pengumuman penawaran publik, *project sign up*, pembukaan, evaluasi dan pengumuman pemenang tender. Sedangkan sistem pengawasan proyek konstruksi (sistem B) berfungsi untuk memajemen deklarasi proyek, pemanggilan untuk menunggu registrasi, kualitas pengawasan,

keamanan dalam pengawasan sampai pada penerimaan akhir. Karena adanya kebijakan pemerintah bahwa semua proses pemanggilan harus melalui sistem A, maka untuk menjamin konsistensi dan keakuratan data, kedua sistem tersebut harus dapat saling bertukar data. Skema pertukaran data yang ditawarkan pada penelitian tersebut menggunakan *web service* yang dibangun dengan menggunakan bahasa pemrograman C#.

Beberapa penelitian yang telah diuraikan diatas kebanyakan menggunakan *SOAP*. Saat ini *SOAP* merupakan rekomendasi *World Wide Web Consortium (W3C)* dan banyak digunakan untuk aplikasi-aplikasi *enterprise* (Refsnes Data, 2011). Namun ada beberapa *web service* lain yang juga sering digunakan seperti *Representational State Transfer (REST)* dan *Extensible Markup Language – Remote Procedure Call (XML-RPC)*. *XML-RPC* dirancang sesederhana mungkin, namun memungkinkan struktur data yang kompleks dikirim, diproses dan dikembalikan (nonename, 2003), sehingga lebih ringan dan mudah diimplementasikan (Allman, 2003). Sampai saat ini banyak *library* yang telah dikembangkan untuk mempermudah implementasi *XML-RPC* pada setiap lingkungan bahasa pemrograman seperti *XML-RPC.NET* untuk .Net (Cook, 2011), *Apache XML-RPC* untuk Java (The Apache Software Foundation, 2010) dan lain-lain. Beberapa penelitian sebelumnya telah menggunakan *XML-RPC* dalam mengimplementasikan sistem terdistribusi, seperti penelitian yang dilakukan oleh Stokes-Rees dkk yang menyajikan pendekatan yang ringan untuk informasi terdistribusi dan sistem monitoring pada paket *software* jaringan *Distributed Implementation with Remote Agent Control (DIRAC)* menggunakan

XML-RPC dan *instant messaging*. Selain itu Tenable Network Security, Inc. juga menggunakan skema *XML-RPC* dalam memberikan akses layanan tertentu yang disediakan oleh *nessus* (*software* untuk monitoring keamanan dari sebuah jaringan) (noname, 2011).

Sistem yang akan dibangun merupakan salah satu bagian yang akan diintegrasikan dengan sistem akademik, yang berfungsi untuk menyediakan beberapa informasi secara mudah dan cepat melalui *SMS*. Sistem ini dapat mengakses basis data akademik melalui *web service* dengan memanggil alias dari *method* yang ada di *server*, dengan parameter tertentu sesuai permintaan yang diperoleh dari *SMS* yang masuk. Sistem ini akan dibangun menggunakan teknologi Java dan integrasinya menggunakan *XML-RPC*. Tabel 2.1 berikut berisi perbandingan penelitian yang akan dilakukan dengan penelitian-penelitian yang telah dilakukan sebelumnya.

Tabel 2.1 Perbandingan Penelitian

| NO | PENELITI | PEMBAHASAN | HUBUNGAN TERHADAP OPERATOR | | ARSITEKTUR SISTEM | |
|----|--|---|----------------------------|------------|-------------------|-----------|
| | | | DEPENDEN | INDEPENDEN | HOMOGEN | HETEROGEN |
| 1 | Cahyo dkk, 2006 | SMS gateway untuk layanan informasi kejuaran sepak bola Liga Inggris yang menggunakan bahasa pemrograman Visual Basic. | | √ | √ | |
| 2 | Mee & Selamat, 2007 | Implementasi SMS gateway pada MLM (Multi Level Marketing). | | √ | √ | |
| 3 | Muhamadi dkk, 2009 | Penambahan fitur SMS gateway pada Student Record Retrieval System untuk memudahkan siswa dalam mengakses informasi terutama dari gurunya secara up to date. | | √ | √ | |
| 4 | Manoj dkk, 2007; Vimal and Chitra, 2008 | SMS gateway untuk layanan administrasi pada perpustakaan. | | √ | √ | |
| 5 | Scomavacca dkk, 2007 | pemanfaatan SMS untuk interaksi tanya jawab dan jajak pendapat di dalam ruang kelas. | | √ | √ | |
| 6 | Balasundaram and Ramadoss, 2007 | Pemanfaatan SMS pada m-Learning. | | √ | √ | |
| 7 | Susanto and Goodwin, 2006; Susanto dkk, 2008 | e-government berbasis SMS. | | √ | √ | |
| 8 | Khiyal dkk, 2009 | SMS gateway untuk mengontrol keamanan rumah : HACS (Home Appliance Control System). | | √ | √ | |
| 9 | Seo dkk, 2007 | Implementasi SMS gateway dalam mekanisme penanganan spam. | | √ | √ | |
| 10 | Calvanese dkk, 2001 | Menggunakan pendekatan deklaratif untuk mengintegrasikan sumber-sumber data pada data warehouse berdasarkan pada formalisme pemodelan konseptual ekspresif yang dilengkapi dengan teknik penalaran, dimana kerangka kerja yang disajikan dalam penelitian tersebut diterapkan untuk mengatasi masalah desain data warehouse dalam proyek ESPRIT DWQ. | | | | √ |
| 11 | Julistiono & Lamidjan, 2001 | Menggunakan teknologi CORBA untuk membuat aplikasi untuk reservasi hotel dengan sistem multi-tier. | | | | √ |
| 12 | Yoon dan Park, 2008 | berbagai bentuk implementasi dari <i>web service SOA (Service-oriented Architecture)</i> untuk memenuhi kebutuhan dalam beradaptasi terhadap keadaan (adaptasi secara dinamis terhadap perubahan frekuensi jumlah pesan <i>XML (Extensible Markup Language)</i> antar berbagai macam situs/aplikasi) dan mempertahankan performance eksekusi (kecepatan dan kelancaran dalam transfer data). | | | | √ |
| 13 | Priyambodo, 2005 | Implementasi web service untuk membangun service yang akan memberikan rekomendasi mengenai sejumlah obyek wisata di Jawa Tengah dan Yogyakarta yang sebaiknya dikunjungi oleh wisatawan dalam rangka wisata budaya Jawa. | | | | √ |
| 14 | Zhang dkk, 2006 | membangun service menggunakan <i>SOAP</i> yang diimplementasikan untuk melakukan pertukaran data antara dua buah sistem manajemen proyek konstruksi. | | | | √ |
| 15 | Stokes-Rees dkk | Menyajikan pendekatan yang ringan untuk informasi terdistribusi dan sistem monitoring pada paket software jaringan <i>DIRAC (Distributed Implementation with Remote Agent Control)</i> menggunakan <i>XML-RPC</i> dan <i>instant messaging</i> . | | | | √ |
| 16 | Tenable Network Security, Inc., 2011 | Tenable Network Security, Inc. menggunakan skema <i>XML-RPC</i> dalam memberikan akses layanan tertentu yang disediakan oleh nessus (software untuk monitoring keamanan dari sebuah jaringan). | | | | √ |
| 17 | Penelitian yang diusulkan | melakukan analisa dan membuat rancang bangun sistem layanan akademik berbasis SMS yang diintegrasikan dengan sistem akademik yang telah ada sebelumnya. sistem yang akan dibangun menggunakan teknologi Java yang lintas platform sehingga dapat diinstall pada berbagai platform komputer tanpa melakukan proses kompilasi berulang kali. Sedangkan integrasinya menggunakan <i>XML-RPC (Extensible Markup Language - Remote Procedure Call)</i> sebagai salah satu teknologi web service yang cukup ringan dan mudah digunakan. | | √ | | √ |

B. Landasan Teori

1. Short message service (SMS)

SMS merupakan fasilitas standar dari *GSM*, walaupun kini sudah banyak pula telepon selular dengan teknologi *Code Division Multiple Access (CDMA)* yang juga dilengkapi dengan fasilitas *SMS*. Fasilitas ini dipakai untuk mengirim dan menerima pesan ke dan dari sebuah telepon selular. (Muhamadi dkk, 2009)

a. Cara kerja SMS

Pada saat pesan *SMS* dikirim dari telepon selular, pesan tersebut tidak langsung dikirim ke telepon selular tujuan, akan tetapi terlebih dahulu dikirim ke *SMS Center* dengan prinsip *store and forward*, setelah itu pesan diteruskan ke *SMS Center* yang lain sesuai operator nomor telepon selular yang dituju dan selanjutnya ke telepon selular yang dituju, atau jika operator telepon selular yang dituju sama dengan pengirim (pesan dikelola oleh *SMS Center* yang sama) maka pesan tersebut dapat langsung diteruskan ke telepon selular yang dituju tanpa melalui *SMS Center* lain (Mulliner and Miller, 2009). Melalui keberadaan *SMS Center* dapat diketahui status dari pesan yang dikirim, apakah telah sampai ataukah gagal diterima oleh telepon selular tujuan. Apabila telepon selular tujuan dalam keadaan aktif dan menerima pesan yang dikirim, maka telepon selular tersebut akan mengirim kembali pesan konfirmasi ke *SMS Center* yang menyatakan bahwa pesan telah diterima. Kemudian *SMS Center* mengirimkan kembali status tersebut kepada si pengirim. Tetapi jika telepon selular tujuan dalam keadaan mati atau di luar jangkauan, maka pesan yang dikirimkan akan disimpan pada *SMS Center* sampai periode validitas terpenuhi, jika periode

validitas terlewat maka pesan itu akan dihapus dari *SMS Center* dan tidak dikirimkan ke telepon selular tujuan. Disamping itu *SMS Center* juga akan mengirim pesan informasi ke nomor pengirim yang menyatakan pesan yang dikirim belum diterima atau gagal.

2. *Protocol data unit (PDU)*

Dalam pengiriman dan penerimaan pesan *SMS*, terdapat dua mode, yaitu mode teks dan mode *Protocol Data Unit (PDU)*. Mode teks adalah format pesan dalam bentuk teks asli yang dituliskan pada saat akan mengirimkan pesan. Sesungguhnya mode teks ini adalah hasil pengkodean dari mode *PDU*. Sedangkan mode *PDU* adalah format pesan dalam bentuk oktet heksadesimal dan oktet semidesimal dengan panjang mencapai 160 (7 bit) atau 140 (8 bit) karakter. Di Indonesia, tidak semua *operator GSM* maupun terminal mendukung mode teks, sehingga mode yang digunakan adalah mode *PDU*. Pada pengiriman pesan terdapat dua jenis *mobile*, yaitu *Mobile Terminated (Telepon selular Penerima)* dan *Mobile Originated (Telepon selular Pengirim)*. (Ortiz and Prieto, 2004)

a. *SMS PDU pengirim*

SMS PDU pengirim adalah pesan yang dikirim dari telepon selular ke *SMS Center*. Pada prinsipnya apabila kita mengirim pesan ke nomor tujuan, pesan itu akan melalui *SMS Center*. Pesan yang akan dikirim oleh terminal masih dalam bentuk teks, sedangkan dalam pengiriman ke *SMS Center* harus dalam bentuk *PDU*. Untuk itu sebelum dikirim, terminal atau telepon selular akan melakukan perubahan dari format teks menjadi *PDU*, proses ini sering disebut proses

encodec. Adapun skema dari format *PDU* pengirim telah diatur dan ditetapkan oleh *ETSI* dapat dilihat pada tabel 2.2 berikut :

Tabel 2.2 Skema Format *SMS PDU* Pengirim (Purnomo, 2005)

| | | | | | | | | |
|-----|-----------------|----|----|-----|-----|----|-----|----|
| SCA | <i>PDU</i> Type | MR | DA | PID | DCS | VP | UDL | UD |
|-----|-----------------|----|----|-----|-----|----|-----|----|

Penjelasan dari masing-masing format adalah sebagai berikut :

1) Service center address (SCA)

SCA adalah informasi dari alamat (nomor) *SMS Center*. *SCA* memiliki tiga komponen antara lain :

- a) *Len*, yaitu panjang informasi *SMS Center* dalam octet.
- b) *Type of number*, yaitu format nomor dari *SMS Center*. Untuk format lokal 81 hexa, sedangkan format internasional 91 hexa.
- c) *Service center number*, yaitu nomor *SMS Center* dari operator pengirim. Jika panjangnya ganjil maka pada karakter terakhir ditambahkan 0F hexa.

Dalam pengiriman *SMS*, nomor *SMS Center* tidak dicantumkan sehingga *SCA* bernilai 00.

2) *PDU* type

Nilai default dari *PDU Type* untuk *SMS* pengirim adalah 11 hexa, yang memiliki arti bahwa 11 hexa = 0010001. Masing-masing mewakili parameter seperti pada tabel 2.3.

Keterangan tabel 2.3 sebagai berikut:

RP : *Reply Path*. Parameter yang menunjukkan bahwa alur jawaban ada.

UDHI : *User Data Header Indicator*. Bit ini bernilai 1 jika data pengirim dimulai dengan suatu judul / tema.

SSR : *Status Report Request*. Bit ini bernilai 1 jika laporan status pengiriman diminta.

VPF : *Validity Period Format*. Format dari batas waktu pengiriman jika pesan gagal diterima.

RD : *Reject Duplicates*. Parameter yang menandakan ya atau tidaknya service center akan menerima suatu pengiriman pesan *SMS* untuk suatu pesan yang masih disimpan dalam service center tersebut.

MTI : *Message Type Indicator*. Bit bernilai 0 untuk menunjukkan bahwa *PDU* ini adalah suatu *SMS-Deliver*.

Tabel 2.3 *PDU* Typc Pengirim (Purnomo, 2005)

| Bit ke | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|----|------|-----|-----|-----|----|-----|-----|
| Nama | RP | UDHI | SRR | VPF | VPF | RD | MTI | MTI |
| Nilai | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |

3) Message reference (MR)

Message Reference adalah acuan dari pengaturan pesan *SMS*. Apabila diberi nilai 00 maka pengaturan pesan *SMS* dilakukan sendiri oleh telepon selular tujuan.

4) Destination address (DA)

Destination Address adalah alamat (nomor) tujuan, yang terdiri atas panjangnya nomor tujuan (Len), format dari nomor tujuan (Type Number) dan nomor tujuan (Destination Number).

5) Protocol identifier (PID)

Protocol Identifier adalah tipe atau format dari cara pengiriman pesan, yang biasanya diatur dari telepon selular pengirim. Misalnya tipe standard text, fax, e-mail, telex, X400 dan lain-lain. Nilai default dari *PID* adalah 00 yang menunjukkan tipe standard text.

6) Data coding scheme (DCS)

Data Coding Scheme adalah rencana dari pengkodean data untuk menentukan kelas dari pesan tersebut, apakah berupa *SMS teks standard*, *flash SMS* atau *blinking SMS*.

7) Validity period (VP)

Validity Period adalah lama waktu pesan *SMS* disimpan di *SMS Center* apabila pesan tersebut gagal diterima oleh telepon selular penerima. Tabel validity period dapat dilihat pada table 2.4.

Tabel 2.4 Validity Period (Purnomo, 2005)

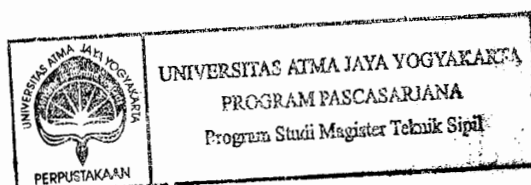
| Waktu VP | Nilai VP |
|-------------------------------|---|
| 5 menit -720 menit (12 jam) | $(\text{Waktu VP} / 5) - 1$ |
| 12,5 jam - 24 jam | $143 + ((\text{Waktu VP} - 12) * 2)$ |
| 2 - 30 hari | $166 + \text{Waktu VP}$ |
| Lebih dari 4 minggu | $192 + \text{Waktu VP}$ |

8) User data length (UDL)

User Data Length adalah panjangnya pesan *SMS* yang akan dikirim dalam bentuk teks standar.

9) User data (UD)

User Data adalah isi pesan yang akan dikirim dalam format hexadesimal.



b. *SMS PDU* penerima

SMS PDU Penerima adalah pesan yang datang atau masuk dari *SMS Center* ke telepon selular dalam format *PDU*. Pada prinsipnya, pesan yang diterima dari *SMS Center* masih dalam format *PDU*, setelah itu terminal / telepon selular yang menerima pesan akan melakukan pengkodean menjadi teks. Proses ini sering disebut proses *decodec*. Format *PDU* dari *SMS* penerima dapat dilihat pada tabel 2.5.

Tabel 2.5 Skema Format *SMS PDU* Penerima (Purnomo, 2005)

| | | | | | | | |
|-----|-----------------|----|-----|-----|------|-----|----|
| SCA | <i>PDU</i> Type | OA | PID | DCS | SCTS | UDL | UD |
|-----|-----------------|----|-----|-----|------|-----|----|

Penjelasan masing-masing format sebagai berikut :

1) Service center address (SCA)

SCA adalah alamat dari *SMS Center*, yang memiliki tiga komponen utama yaitu *len*, *type of number* dan *service center number*.

2) PDU type

Nilai default dari *PDU Type* untuk *SMS Deliver* adalah 04 hexa, yang memiliki arti 04 hexa = 00000100. Rormat *PDU* type penerima dapat dilihat pada tabel 2.6.

Keterangan table 2.6 :

RP : *Reply Path*. Parameter yang menunjukkan bahwa alur jawaban ada.

UDHI : *User Data Header Indicator*. Bit ini bernilai 1 jika data pengirim dimulai dengan suatu judul / tema.

SRI : *Status Report Indication*. Bit ini bernilai 1 jika suatu status laporan akan dikembalikan ke *SMS*.

MMS : *More Message to Send*. Bit ini bernilai 0 jika ada pesan lebih yang akan dikirim.

MTI : *Message Type Indicator*. Bit bernilai 0 untuk menunjukkan bahwa *PDU* ini adalah suatu *SMS-Deliver*.

Tabel 2.6 *PDU* Type Penerima (Purnomo, 2005)

| | | | | | | | | |
|--------|----|------|-----|------|------|-----|-----|-----|
| Bit ke | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Nama | RP | UDHI | SRI | <nn> | <nn> | MMS | MTI | MTI |
| Nilai | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

3) Originator address (OA)

OA adalah alamat dari pengirim, yang terdiri atas panjangnya nomor pengirim (*len*), format dari nomor pengirim (*type number*) dan nomor pengirim (*originator number*).

4) Protocol identifier (PID)

PID adalah tipe atau format dari cara pengiriman pesan, yang biasanya diatur dari telepon selular pengirim. Misalnya tipe standard teks, fax, e-mail, telex, X400 dan lain-lain. Nilai default dari *PID* adalah 00 yang menunjukkan tipe standard text.

5) Data coding scheme (DCS)

Data Coding Scheme adalah rencana dari pengkodean data untuk menentukan kelas dari pesan tersebut, apakah berupa *SMS standard teks*, *flash SMS* atau *blinking SMS*.

6) Service center time stamp (SCTS)

Service Center Time Stamp adalah waktu dari penerimaan pesan oleh SMS Center penerima. *SCTS* terdiri atas tahun, bulan, tanggal, jam, menit dan detik, serta zona waktu.

7) User data length (UDL)

User Data Length adalah panjang dari pesan yang diterima dalam bentuk teks standar.

8) User data

User Data adalah pesan yang diterima dalam format hexadesimal.

3. AT command

Attention Terminal Command (AT Command) adalah perintah-perintah yang digunakan untuk berkomunikasi dengan terminal (telepon selular) melalui serial port pada komputer. Dengan *AT Command*, dapat diketahui vendor dari telepon selular yang digunakan, kekuatan sinyal, membaca pesan yang ada pada *SIM (Subscriber Identity Module) card* dan lainnya. Tabel 2.7 berisi perintah-perintah *AT* yang terkait dengan SMS Server.

Dalam menggunakan *AT Command* ada beberapa hal yang harus diperhatikan yaitu *command* apa yang harus dimasukan pada terminal, tindakan apa yang harus dilakukan setelah *command* dimasukan ke terminal dan mengetahui respon apa yang didapat setelah melakukan tindakan pada *command* yang dimasukan.

Tabel 2.7 AT Command yang Digunakan pada SMS Server (Purnomo, 2005)

| AT Command | Keterangan |
|------------|---|
| AT | Mengecek apakah telepon selular telah terhubung |
| AT + CMGF | Menetapkan format mode dari terminal |
| AT + CSCS | Menetapkan jenis encoding |
| AT + CNMI | Mendeteksi pesan SMS baru masuk secara otomatis |
| AT + CMGL | Membuka daftar SMS yang ada pada SIM Card |
| AT + CMGS | Mengirim pesan SMS |
| AT + CMGR | Membaca pesan SMS |
| AT + CMGD | Menghapus pesan SMS |

4. Sistem informasi

Menurut Oetomo (2002), sistem informasi merupakan kumpulan elemen yang saling berhubungan satu sama lain yang membentuk satu kesatuan untuk mengintegrasikan data, memproses, menyimpan, dan mendistribusikan informasi. Keluaran dari sistem informasi adalah informasi, yang dapat dipakai oleh setiap individu dan aplikasi, baik dalam organisasi maupun mitra organisasi sesuai kepentingannya masing-masing.

Sistem informasi memiliki peran yang cukup penting dalam membantu suatu organisasi meningkatkan produktifitas, mengurangi biaya-biaya tertentu, meningkatkan pelayanan terhadap konsumen dan pada tingkat manajerial dapat membantu dalam pengambilan keputusan (Budiman and Iswanti). Selain itu sistem informasi juga dapat menyediakan komunikasi dalam dan antar organisasi secara akurat dan cepat. Beberapa hal tersebut menjadi karakteristik sistem

informasi yang harus dipenuhi dan harus diperhatikan ketika merancang sebuah sistem informasi.

5. SMS gateway

Dengan adanya *AT Command* sebagai jembatan komunikasi antara telepon selular dan komputer, menjadikan informasi lebih leluasa untuk disampaikan secara otomatis melalui berbagai media. *SMS gateway* merupakan mekanisme mengirim dan menerima pesan singkat berupa teks melalui sebuah komputer yang terhubung ke telepon selular atau *modem GSM* melalui *serial port*, *IrDA* (*Infrared Data Association*) *port* maupun *bluetooth*. Dimana telepon selular maupun *modem GSM* berfungsi sebagai *modem*. Arsitektur ini disebut *independent service*. Arsitektur lain untuk menghubungkan antara penerima dan penyedia informasi melalui *SMS* yaitu *dependent service*, dimana komputer yang berfungsi sebagai *server gateway* terhubung secara langsung ke server operator selular melalui internet (Pramsane & Sanjaya, 2006).

Dengan aplikasi *SMS gateway*, *SMS* dapat dimanfaatkan untuk keperluan yang lebih luas dalam menyediakan informasi sejenis bagi banyak orang sesuai permintaan dengan format tertentu secara otomatis. *SMS gateway* telah diimplementasikan pada berbagai bidang terutama di dunia bisnis. Dalam bidang pendidikan, *SMS gateway* dapat dimanfaatkan untuk melayani permintaan informasi nilai, jadwal kuliah, jadwal ujian, absensi kehadiran, tunggakan SPP dan lain sebagainya. Informasi ini juga dapat dikirimkan ke orang tua/wali sehingga dapat membantu orang tua dalam mengawasi anaknya di lembaga pendidikan. *SMS gateway* juga dapat dimanfaatkan sebagai *SMS reminder* yang

digunakan untuk mengingatkan mahasiswa terkait proses bimbingan penelitian Kerja Praktek dan Tugas Akhir serta melaksanakan proses pelaporan kemajuan (*progress report*) pelaksanaan penelitian Tugas Akhir (Hendrik, 2007).

6. Integrasi aplikasi

Integrasi aplikasi dapat diartikan berbagi pakai data dan fungsi tanpa batas diantara semua aplikasi dan basis data yang saling terhubung. Fajar (2009) menyatakan bahwa : “enterprise application integration memadukan aplikasi dan sumber data enterprise sehingga dengan mudah mereka dapat berbagi bisnis proses dan data”. Secara umum integrasi dapat dilakukan pada level data maupun level logic.

Integrasi pada level data meliputi perpindahan data antar data stores dan penggunaan data secara bersama-sama untuk sharing informasi antar aplikasi yang ada. Integrasi pada level data dilakukan tanpa membuat perubahan signifikan pada aplikasi. Teknologi yang dapat digunakan antara lain : *middleware* yang dapat dibangun menggunakan *Open Data Base Connectivity (ODBC)*, *Java Data Base Connectivity (JDBC)* atau *Application Programming Interface (API)* lainnya, *data warehouse* seperti *Extract Transform Load (ETL)* dan lain-lain.

Integrasi pada level logic terfokus pada *method (function/procedure)*. *Method* di-shared dengan meletakkannya pada sebuah server pusat (*application server*) atau dengan mengakses *method* antar aplikasi (*distributed object*). Beberapa teknologi yang dapat digunakan untuk melakukan integrasi pada level ini antara lain : *procedural middleware* seperti *Remote Procedure Call (RPC)* yang memberikan mekanisme pemanggilan sebuah fungsi yang secara aktual berada di

program lain pada mesin lain. Selain itu ada beberapa teknologi lain seperti *Common Object Request Broker Architecture (CORBA)*, *Distributed Component Object Model (DCOM)* dan *Java Remote Method Invocation (Java RMI)*, dimana ketiganya memiliki prinsip kerja yang sama yaitu aplikasi-aplikasi yang mengimplementasikannya harus menggunakan standar *interface* dan *protocol* yang sama, yang telah disepakati untuk saling berkomunikasi antar aplikasi (disebut *distributed object*). Perbedaan ketiganya terletak pada batasannya, dimana *CORBA* mendukung komunikasi antar aplikasi dengan bahasa pemrograman yang berbeda, lintas *platform* dan memiliki banyak fitur; *DCOM* mendukung beberapa bahasa pemrograman dalam lingkungan Windows; sedangkan *Java RMI* hanya mendukung bahasa pemrograman Java.

Teknologi berbasis komponen seperti *CORBA*, *DCOM* dan *Java RMI* berorientasi pada koneksi dan tidak menggunakan *port* komunikasi standar, untuk menjaga hak akses terkait isu keamanan. Sehingga dibutuhkan konfigurasi khusus di sisi *firewall* dan cukup sulit diimplementasikan. Teknologi juga ini tidak dapat menangani masalah gangguan dengan baik karena biaya yang terlalu tinggi untuk menghubungkan kembali ke *remote server* setelah terjadi perubahan. Seiring dengan berkembangnya internet, berkembang pula teknologi *web service* yang menjadikan integrasi sistem semakin mudah dilakukan, salah satunya karena menggunakan *port 80* sebagai *port* standar dalam pertukaran data (Shetty and Vadivel, 2009).

7. Web service

Menurut Kreger (2001) *web service* diartikan sebagai sebuah antar muka (*interface*) yang menggambarkan sekumpulan operasi-operasi yang dapat diakses melalui jaringan, misalnya internet, dalam bentuk pesan *XML*. Secara sederhana, *web service* dapat didefinisikan sebagai sebuah fungsi (*method*) dari aplikasi yang menyediakan sebuah *API* bagi aplikasi lain yang dapat dipanggil dengan aman melalui web dengan protokol *Hyper Text Transfer Protocol (HTTP)*, tanpa memerlukan konfigurasi khusus di sisi *firewall*. *Web service* dapat dibangun dengan menggunakan bahasa pemrograman apa saja dan dapat diimplementasikan pada berbagai platform, sehingga memungkinkan komunikasi antar berbagai aplikasi yang berbeda, baik dari sisi bahasa pemrograman yang digunakan dalam membuat aplikasi tersebut, sistem operasi maupun hardware tempat aplikasi tersebut berjalan (Zhang dkk, 2006).

Model dari sebuah *web service* didasarkan pada interaksi antara 3 komponen yang berperan dalam *web service*, yaitu: *service provider*, *service registry* dan *service requestor/consumer*. Interaksi yang terjadi antara ketiga komponen tersebut juga melibatkan operasi *publish*, *find* dan *bind*. *Service provider* menyediakan service yang dapat diakses melalui jaringan komputer, misalnya internet. Kemudian, *service provider* mendeskripsikan service yang dibangun dan mempublikasikan *service description* tersebut ke *service registry* atau secara langsung ke *service requestor/consumer*. *Service requestor/consumer* menggunakan operasi *find* untuk mendapatkan *service description* secara lokal maupun melalui *service registry*. *Service description* yang diperoleh itu kemudian

digunakan untuk men-*bind service provider* dan berinteraksi dengan implementasi *web service* yang akan digunakan tersebut (Priyambodo, 2005).

Interaksi antara ketiga komponen seperti yang telah diuraikan diatas dapat terjadi karena adanya kerja sama antara beberapa komponen sebagai standar komunikasi *web service* seperti *HTTP*, *XML*, protokol pertukaran data *REST (Representational State Transfer)*, *SOAP (Simple Object Access Protocol)*, *XML-RPC* dan *Java Script Object Notation - Remote Procedure Call (JSON-RPC)*), *WSDL*, *UDDI* dan lain-lain. *HTTP* adalah *protocol* komunikasi pada layer aplikasi. *XML* merupakan format standar yang digunakan untuk komunikasi antar sistem. Protokol pertukaran data merupakan protokol untuk pertukaran pesan (data/informasi) dalam format *XML* atau *JSON*. *Web Service Description Language (WSDL)* yaitu format yang digunakan untuk mendeskripsikan argumen input/output. *WSDL* adalah bahasa berbasis *XML* yang digunakan untuk mendefinisikan *web service* dan menggambarkan bagaimana cara untuk mengakses *web service* tersebut. Dan *UDDI (Unified Description Discovery Integrated)* merupakan cara untuk mempublikasikan dan menemukan informasi tentang *web service* yang terdaftar (Zhang dkk, 2006; Priyambodo, 2005; nonename, 2010).

Berikut cara kerja *web service* : untuk mencari sebuah *web service*, *service consumer* terlebih dahulu mengirimkan *query* ke *UDDI registry* untuk mencari *service* yang diinginkan. Dari *tModel*, *service consumer* akan mendapatkan deskripsi *WSDL* yang menyatakan antarmuka *service (service interface)*. Dengan menggunakan deskripsi *WSDL* yang telah diperoleh tersebut, *service consumer*

dapat membuat sebuah pesan yang sesuai dengan jenis protokol *web service* yang digunakan, untuk berkomunikasi dengan *web service* yang diinginkan.

